



Betriebssysteme und Sicherheit, WS 2020/21

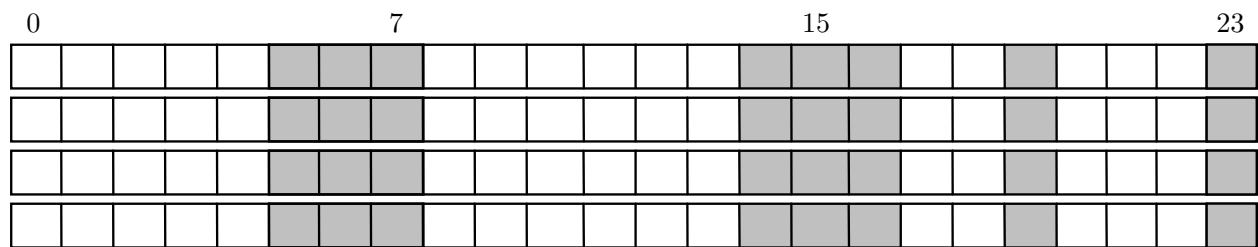
5. Aufgabenblatt – Speicherverwaltung

Geplante Bearbeitungszeit: drei Wochen

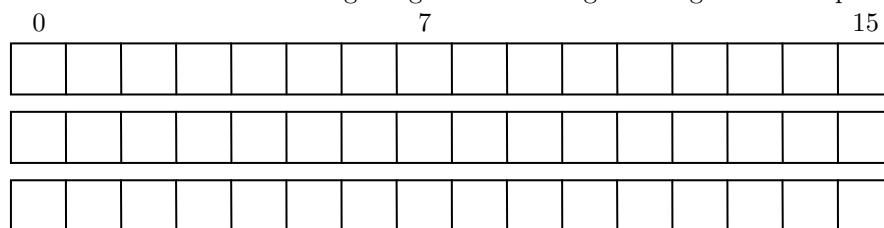
Aufgabe 5.1 Eine Möglichkeit, mehrere Programme gleichzeitig laufen zu lassen, ist die Einteilung des Speichers in feste Partitionen. Erklären Sie die damit verbundenen Probleme und ihre Lösung.

Aufgabe 5.2 Der Übergang von festen zu variablen Partitionen (Segmenten) erfordert eine entsprechende Verwaltung. Dazu werden im wesentlichen – unter Nutzung von Listen oder Bitmaps – verschiedene Einlagerungsstrategien oder das Buddy-Verfahren verwendet (siehe auch TANENBAUM, Kap. 3.2). Erläutern Sie die generellen Vor- und Nachteile dieser Vorgehensweise. Erklären und bewerten Sie die Einlagerungsstrategien an folgendem Beispiel:

- HS-Gesamtkapazität: 24 KiByte
- Anfangsbelegung (belegte Blöcke in KiByte): 5-7 / 14-16 / 19 / 23
- nacheinander sollen Segmente der Größe 3 – 5 – 1 – 5 (in KiByte) eingelagert werden.



Aufgabe 5.3 Diskutieren Sie das Buddy-Verfahren (Vorgehen, Vorteile und Probleme) anhand des folgenden Beispiels. Ein Speicher kann 16 Seiten aufnehmen. Nacheinander sind Segmente A, . . . , F einzulagern, die aus 5 – 1 – 3 – 4 – 2 – 3 Seiten bestehen (kann ein Segment bei seiner Anforderung nicht eingelagert werden, so wird es für den nächstmöglichen Zeitpunkt vorgemerkt). Danach wird Segment E und schließlich Segment B freigegeben. Geben Sie den Ablauf der Einlagerungen und die Lage der Segmente im Speicher an.



Aufgabe 5.4 Geben Sie einen Überblick über den Begriff „virtueller Speicher“ (Anliegen, Aufgaben, Begriffe, Vorgehensweise, Vor- und Nachteile). Verdeutlichen Sie das Problem von Platzbedarf und Tabellenauslastung bei der Verwendung einfacher Seitentabellen anhand des nachfolgenden Beispiels, und diskutieren Sie Lösungen für dieses Problem.

- zugrundeliegende Architektur: 64 Bit
- Größe des physischen Speichers: 256 MiByte
- Seitengröße: 8 KiByte
- Größe eines Eintrags in der Seitentabelle: 8 Byte.

Aufgabe 5.5 Betrachtet werde ein System mit folgenden Eigenschaften:

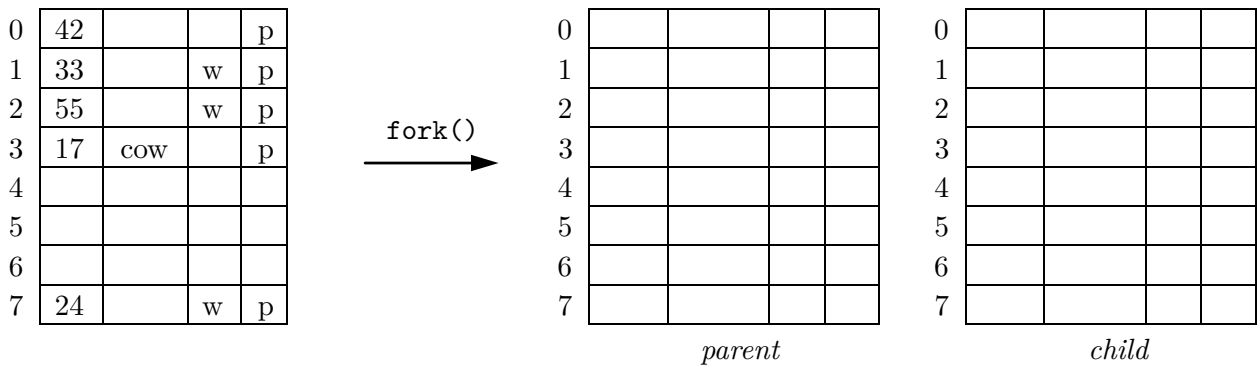
- Seitengröße: 4 KiByte
- Größe des virtuellen Speichers: 64 KiByte
- Größe des realen Speichers: 32 KiByte.

Geben Sie sowohl die einfachen Seitentabellen als auch die invertierte Seitentabelle für folgenden Systemzustand an:

- In Prozess P_1 bestehen folgende Abbildungen von virtuellen auf physische Adressen:
 $0x13AF$ auf $0x53AF$, $0x2745$ auf $0x3745$, $0xF30D$ auf $0x30D$.
- In Prozess P_2 gelten folgende Abbildungen:
 $0x1ABD$ auf $0x4ABD$, $0x2007$ auf $0x6007$, $0x3FFE$ auf $0x7FFE$.

Was ändert sich, wenn in Prozess P_2 die virtuelle Adresse $0x1ABD$ auf $0x5ABD$ statt auf $0x4ABD$ abgebildet werden soll?

Aufgabe 5.6 Zum Erzeugen eines neuen Prozesses wird in Unix der Systemruf `fork()` benutzt. Der Adressraum des dadurch erzeugten Prozesses ist eine identische Kopie des Adressraums des erzeugenden Prozesses. Da in der Regel der neue Prozess sofort ein `execve()` ausführt und dabei den Inhalt seines Adressraums ersetzt, versucht man das tatsächliche Kopieren zu vermeiden. Die zugehörige Technik heißt copy on write. Erläutern Sie, wie copy on write prinzipiell funktioniert, und führen Sie es für den folgenden, durch eine einfache Seitentabelle beschriebenen Adressraum exemplarisch durch (die 2. Spalte der Tabelle werde dabei für das COW-Bit genutzt). Was geschieht, wenn nach dem Kopieren zunächst der Kind-Prozess und danach der Vater-Prozess schreibend auf Seite 7 zugreifen?



Aufgabe 5.7 Erläutern und bewerten Sie die verschiedenen Strategien zur Verdrängung (Ersetzung) von Seiten in einem System mit virtuellem Speicher anhand der folgenden Beispiele. Dabei wird, sofern nicht anders angegeben, jeweils ein physischer Speicher mit vier Rahmen/Kacheln zugrunde gelegt.

- (a) Für die Seitenreferenzfolge 1 – 2 – 3 – 4 – 1 – 2 – 5 – 1 – 2 – 3 – 4 – 5 sind die optimale Seitenersetzung sowie die Strategien FIFO und LRU zu betrachten. Bestimmen Sie zusätzlich für den Algorithmus FIFO die Anzahl der Seitenfehler in dem Fall, dass nur drei Rahmen vorhanden sind. Welches „unnormale“ Verhalten bei FIFO zeigt das Beispiel, wenn die Anzahl der Rahmen vergrößert wird?

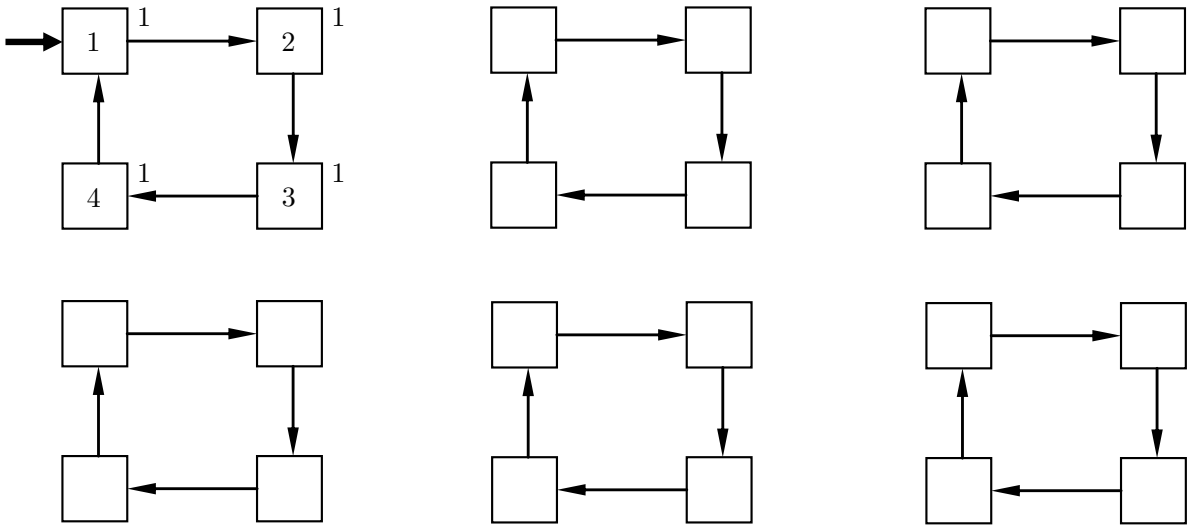
OPT	1	2	3	4	1	2	5	1	2	3	4	5
Rahmen 1												
2												
3												
4												
Seitenfehler												

FIFO	1	2	3	4	1	2	5	1	2	3	4	5
Rahmen 1												
2												
3												
4												
Seitenfehler												

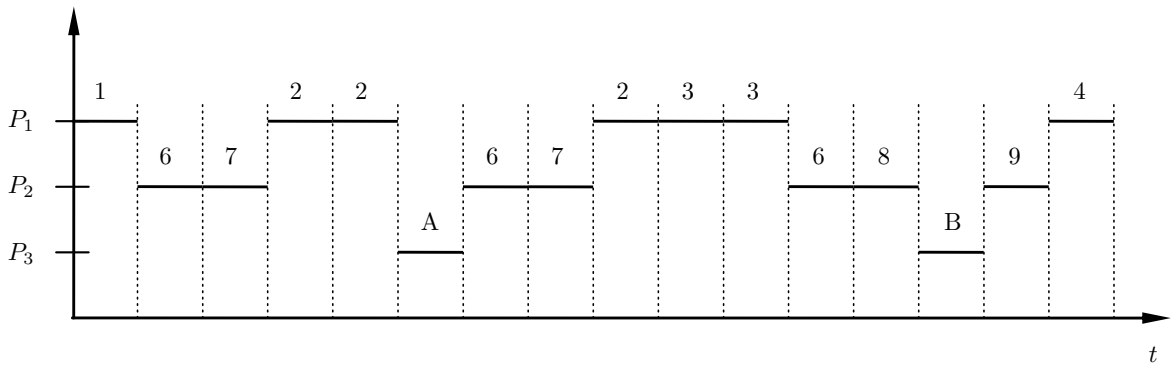
FIFO	1	2	3	4	1	2	5	1	2	3	4	5
Rahmen 1												
2												
3												
Seitenfehler												

LRU	1	2	3	4	1	2	5	1	2	3	4	5
Rahmen 1												
2												
3												
4												
Seitenfehler												

(b) Eine Variante von FIFO, die auch das Referenzverhalten der Prozesse berücksichtigt, ist der Second-Chance- bzw. Clock-Algorithmus. Diskutieren Sie diesen Algorithmus für die Referenzfolge 1 – 2 – 3 – 4 – 2 – 3 – 5 – 2 – 3 – 1 – 2 – 4.



(c) Aufgrund der Seitenreferenzen dreier Prozesse möge sich die in unten stehender Abbildung dargestellte resultierende Referenzfolge ergeben. Erläutern Sie an diesem Beispiel den Begriff „Arbeitsmenge“. Tragen Sie in der Tabelle die entstehende Speicherbelegung ein unter der Annahme, dass fünf Rahmen verfügbar sind und dass jeder Prozess einen Arbeitsmengenparameter (Fenstergröße) von 2 Seiten besitzt. Erklären Sie ferner den Thrashing-Effekt.

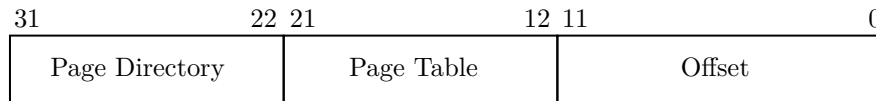


Rahmen	1	6	7	2	2	A	6	7	2	3	3	6	8	B	9	4
1																
2																
3																
4																
5																

Hinweis:

Die restlichen Aufgaben basieren auf der x86-Prozessor-Familie und gehen daher von folgenden Voraussetzungen aus:

- 32-Bit-Adressen
- Größe des virtuellen Adressraums: 4 GiByte
- Seitengröße: 4 KiByte
- zweistufige Seitentabellen: die ersten 10 Bit der Adresse dienen als Index für die erste Stufe (dem Seitenverzeichnis, Page Directory), die nächsten 10 Bit als Index für die zweite Stufe (den Seitentabellen, Page Table) und der Rest als Offset innerhalb der Seite:



- ein Eintrag in einer Seitentabelle hat die folgende Struktur:



Bedeutung der Einträge (jeweils bei gesetztem Bit):

- P Present Bit – die zum Eintrag gehörende Seite ist im Speicher
- W Write Bit – die Seite ist auch schreibbar
- U User Bit – die Seite ist für den Nutzer zugreifbar
- A Accessed Bit – auf die Seite wurde zugegriffen
- D Dirty Bit – die Seite wurde verändert

Die anderen Bits werden hier nicht näher betrachtet.

Aufgabe 5.8 Gegeben sei der durch die abgebildete zweistufige Seitentabelle beschriebene Adressraum, wobei 0x1000 die Eintrittsadresse für das Page Directory ist. Alle angegebenen Zahlen sind Hexadezimal notiert, ein freies Feld in den letzten drei Spalten bedeutet, dass das entsprechende Bit nicht gesetzt ist. Betrachtet werde eine Menge von Speicherzugriffen aus dem normalen „user mode“ heraus (siehe Tabelle, Bedeutung der Abkürzungen siehe Aufgabe 8). Entscheiden Sie für jeden Speicherzugriff, ob der Zugriff gestattet ist und geben Sie in diesem Fall die physische Adresse an. Bestimmen Sie andernfalls den Grund für auftretende Seitenfehler und beschreiben Sie die Reaktion des Betriebssystems.

virt. Adresse	Zugriffsart	PD	PT	physische Adresse bzw. Grund für PF
0x1234	lesend			
0x42D8E4	lesend			
0x1256	schreibend			
0x400985	schreibend			
0x7DF87	schreibend			
0x940AFE	lesend			
0xC03FEB81	lesend			

1000

0	C		u	w	p
1	57		u		p
2	12		u	w	
...					
300	2			w	p
...					

2000

0	0			w	p
1	1			w	p
...					
3FE	3FE			w	p
3FF	3FF			w	p

C000

0	13		u		p
1	123		u		p
...					
7C	89		u	w	p
7D	9234		u	w	

12000

0	FE		u	w	p
1	D78		u	w	p
...					
140	A23		u	w	p
141	345		u	w	p

57000

0	3		u	w	p
1	81		u	w	p
...					
2D	96		u	w	p
2E	134		u	w	p

Aufgabe 5.9 Im Unix-Teil der Vorlesung wurde die logische Struktur des Adressraums eines Unix-Prozesses vorgestellt. Die Bereiche innerhalb dieses Adressraums werden durch das Betriebssystem mittels geeigneter Datenstrukturen – analog zum Prozentssteuerblock – verwaltet.

- (a) Geben Sie einen möglichen Aufbau einer solchen Datenstruktur in Form einer Tabelle an, die die Informationen enthält, die zur Beschreibung der einzelnen Bereiche erforderlich sind.
- (b) Bilden Sie die folgende Situation mit Hilfe dieser Datenstruktur ab: Die Größe des Text-, Daten- und BSS-Segments eines Prozesses betrage 20 KiByte, 6 KiByte bzw. 11 KiByte (jeweils aufgerundet). Ferner habe der Prozess während seiner Laufzeit ab Adresse 0x5009000 einen 24 KiByte großen Ausschnitt aus einer Datei schreibbar eingeblendet, beginnend ab dem Datei-Offset 0x7000. Der Stack (Keller) beansprucht aktuell eine Seite.

- (c) In diesem Adressraum sei auf folgende Adressen erfolgreich lesend zugegriffen worden und es gelte die nebenstehende Rahmenzuordnung. Schlagen Sie für die fehlenden Einträge geeignete Rahmen vor und konstruieren Sie die entsprechende einfache Seitentabelle.

virtuelle Adresse	physische Adresse
0x1000	0x340000
0x7000	0x32000
0x8000	
0xA000	
0x5009000	0x13000
0xBFFFF000	0x42000
0xC0000000	0
0xC0001000	0x1000

- (d) Was geschieht bei einem Lesezugriff auf die Adressen 0x3400, 0x500D100, 0xC0000040 und einem Schreibzugriff auf 0xA780, 0x3B060, 0xBFFEDCB, wenn der Zugriff von einem normalen Nutzerprogramm aus erfolgt?

Klausuraufgabe I

Ein System verwendet das Buddy-Verfahren zur Verwaltung des Speichers. Der freie Speicher umfasst 16 Seiten. Es werden nacheinander 5 Segmente (A–E), welche jeweils aus 1, 7, 3, 3 und 2 Seiten bestehen in den Speicher eingelagert. Kann eine Anfrage nicht erfüllt werden, so wird diese Anfrage ignoriert und mit der folgenden Anforderung fortgefahren.

Geben Sie den Ablauf der Einlagerungen und die Lage der Segmente im Speicher an. Kennzeichnen Sie nicht erfüllbare Anfragen.

A:1

B:7

C:3

D:3

E:2

Klausuraufgabe II

Betrachtet wird ein System, dessen virtueller und physischer Adressraum jeweils mit 32 Bit breiten Adressen angesprochen wird. Die Seitengröße beträgt 4 KiB, die Seitentabelle ist zweistufig. Die einzelnen Tabellen benötigen jeweils 4 KiB Speicherplatz und enthalten jeweils 1024 Einträge.

HINWEIS: Die Einheit KiB bezeichnet 1024 Byte.

- a) Wie viele Einträge werden in beiden Stufen benötigt um den gesamten virtuellen Adressraum abzubilden?

Eine Anwendung benötigt ein 11 KiB großes Code-Segment und ein 25 KiB großes Datensegment. Das Datensegment soll schreibbar, das Code-Segment nicht schreibbar sein. Das Betriebssystem verwaltet Speicherbelegungen nur in ganzen Seiten, ansonsten dürfen die Segmente beliebig im virtuellen Adressraum platziert werden.

- b) Wie viel physischer Speicher wird im günstigsten und im ungünstigsten Fall benötigt? Berücksichtigen Sie sowohl den Speicherbedarf der Anwendung selbst als auch den Speicher, der für die Seitentabellen erforderlich ist.

Klausuraufgabe III

Gegeben seien untenstehende Speicherverwaltungsstrukturen eines 32-bit Systems mit zweistufigen Seitentabellen. Es werden 10 Bit für die Indizierung in das Seitenverzeichnis (SV) verwendet, sowie 10 Bit für die Indizierung in die Seitentabelle (ST). Der Offset beträgt 12 Bit.

0x0	0x01	w
0x1	0x76	wp
...
0xBF	0x3F	wp
...
0x2FC	0x76	wp
...
0x2FF	0x4F	wp
...

0x0	0x0	wp
0x1	0x0	p
...
0x3FE	0xFFE	wp
0x3FF	0xFFE	wp

0x0	0x123	
0x1	0x0	p
...
0x3FE	0x2A	
0x3FF	0xFFC	wp

0x0	0x0	wp
0x1	0x0	p
...
0x3FE	0x1A	
0x3FF	0xFFD	wp

- a) Geben Sie, sofern möglich, die physischen Adressen der lesenden Zugriffe auf folgende virtuelle Adressen an. Falls ein Seitenfehler auftritt, nennen Sie den Grund dafür.
- 0x400FFD:
 - 0x760DE:
 - 0xBFFFFFF02:
 - 0x2FC00000:
- b) Skizzieren Sie in groben Zügen, was passiert, wenn ein schreibender Zugriff auf die virtuelle Adresse 0xBFFFFFF8 erfolgt? Der Stack reicht aktuell von 0xBFFFFFF000 bis 0xBFFFFFFF und wächst nach unten.

Ein 64-bit System verwendet vierstufige Seitentabellen zur Adressierung des kompletten virtuellen Adressraums. Die Tabellen der verschiedenen Stufen enthalten jeweils 4096 Einträge und füllen jeweils einen vollständigen Rahmen.

- c) Wie viele Bits der Adresse bilden den Offset?
- d) Wie groß ist eine Seite?
- e) Wie groß ist ein einzelner Seitentabelleneintrag?

Klausuraufgabe IV

Gegeben sei eine 32-Bit Architektur mit virtuellem, byteadressierbarem Speicher und zweistufigen Seitentabellen. Das Seitenverzeichnis (SV) und die Seitentabellen (ST) enthalten jeweils 1024 Einträge, welche jeweils 4 Byte groß sind. Der Offset beträgt 12 Bit.

- a) Wie groß ist der virtuelle Adressraum? Wie groß ist eine Seite?

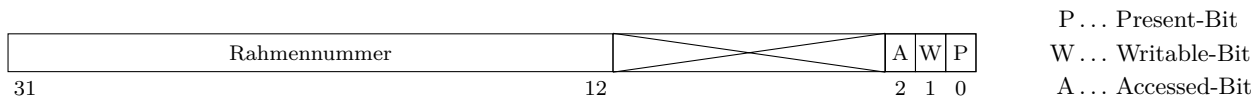
Im folgenden sei eine Seitentabellenstruktur gegeben. Alle nicht genannten oder frei gelassenen Einträge werden als ungültig angenommen.

0x0	0x8007
...	...
0x300	0x4007
...	...

...	...
0x100	0xFFFF0007
...	...
0x2FF	0xB1F006
...	...

...	...
0x20	0xC0100004
0x21	0xA005
...	...
0x2FF	0xB1F007
...	...
0x3FF	0x7AC003

Seitentabellen- und Seitenverzeichniseinträge haben dabei folgenden Aufbau:

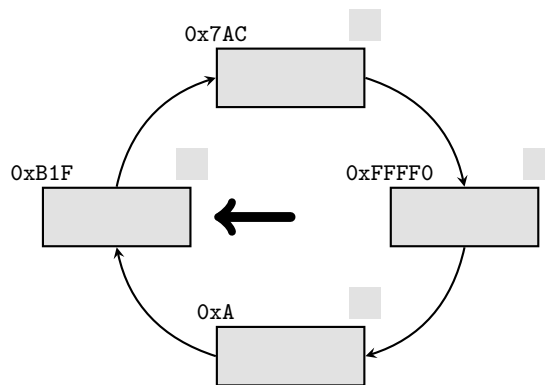


b) Nennen Sie die *virtuellen Adressen*, die laut obiger Seitentabellen auf folgende *physische Adressen* abgebildet werden:

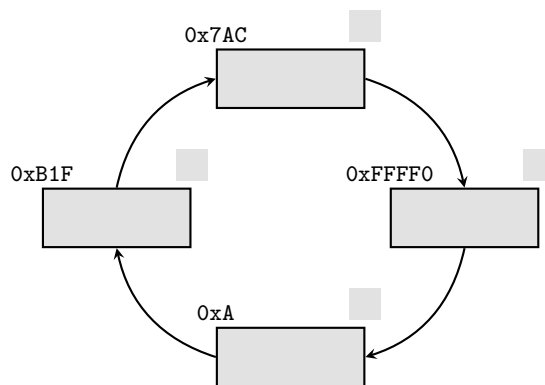
- 0xAFFE:
- 0xFFFF0124:
- 0x7ACE01:
- 0xB1FABC:

c) Für die in b) verwendeten vier Rahmen kommt der Clock-Algorithmus als Seitenersetzungsstrategie zum Einsatz. Ergänzen Sie in den unten stehenden Schemata jeweils die Seiten, die den genannten Rahmen zugeordnet sind, die Accessed-Bits sowie den Clock-Zeiger, so dass die folgenden Situationen korrekt wiedergegeben werden. HINWEIS: Sie können die Seitennummer, also die virtuellen Adressen ohne den Offset, anstelle der kompletten virtuellen Adresse angeben. Alle auftretenden Speicherzugriffe sind zulässig.

I. Tragen Sie zunächst die in b) vorliegende Ausgangssituation ein.



II. Nun erfolgt ein schreibender Zugriff auf die virtuelle Adresse 0x40042. Tragen Sie den Zustand des Clock-Algorithmus nach diesem Zugriff ein.



III. Nun findet ein lesender Zugriff auf die virtuelle Adresse 0x2FF123 statt, gefolgt von einem lesenden Zugriff auf die virtuelle Adresse 0x3FF137. Ergänzen Sie wiederum die Darstellung des Clock-Algorithmus.

